

PATENT

Docket No. END920050015US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

INVENTORS: Chua, Sook C.

APPLICATION NO. 10/804,644

Confirmation No. 7732

FILED: March 19, 2004

Examiner: Cloud, J.

CASE NO. END920050015US1

Group Art Unit: 2144

TITLE: J2EE APPLICATION VERSIONING STRATEGY

FILED ELECTRONICALLY ON August 24, 2009

Commissioner for Patents
MAIL STOP APPEAL BRIEF-PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

Attention: Board of Patent Appeals and Interferences

AMENDED APPELLANTS' BRIEF

This amended brief is being filed in response to the Notification of Non-Compliant Appeal Brief, dated July 23, 2009. The original brief was filed in furtherance of the Notice of Appeal filed in this case on March 1, 2009. The Commissioner was authorized to charge the fee for filing of this Appeal Brief to Deposit Account No. 09-0457 when the original Appeal Brief was filed on April 30, 2009.

1. REAL PARTY IN INTEREST

The present application is assigned to International Business Machines Corporation, having its principal place of business at New Orchard Road, Armonk, New York 10504. Accordingly, International Business Machines Corporation is the real party in interest.

2. RELATED APPEALS AND INTERFERENCES

The appellant, assignee, and the legal representatives of both are unaware of any other appeal or interference which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

3. STATUS OF CLAIMS

- A. Claims canceled: 3, 10 and 17
- B. Claims withdrawn from consideration but not canceled: None
- C. Claims pending: 1-2, 4-9, 11-16, and 18-22
- D. Claims allowed: none
- E. Claims rejected: 1-2, 4-9, 11-16, and 18-22
- F. Claims appealed: 1-2, 4-9, 11-16, and 18-22

Appealed claims 1-2, 4-9, 11-16, and 18-22 as currently pending are attached as the Claims Appendix hereto.

4. STATUS OF AMENDMENTS

No amendments were made in Applicant's Reply, dated December 2, 2008, filed in response to the final Office Action mailed October 2, 2008. The submission of the Reply did not result in allowance by the Examiner.

5. SUMMARY OF THE CLAIMED SUBJECT MATTER

Claim 1: A method for managing the invocation of multiple versions of a J2EE program, stored on an application server, among multiple clients accessing the application server, using an identical service name for the invocation of the multiple versions of the J2EE program (*paragraph [0009]*), comprising: interposing a JNDI proxy between each client and the application server (*paragraph [0009], Fig. 3*); associating each client with one of said versions (*paragraphs [0009] and [0026]; Fig. 7, step 706*); and using said JNDI proxy, directing the version associated with a particular client to said particular client upon a request by said particular client for said J2EE program (*Fig. 7, steps 710-714; paragraphs [0031]-[0032]*).

Claim 8: A system for managing the invocation of multiple versions of a J2EE program, stored on an application server, among multiple clients accessing the application server, using an identical service name for the invocation of the multiple versions of the J2EE program (*paragraph [0009]*), comprising: a JNDI proxy interposed between each client and the application server (*paragraph [0009]; Fig. 3*); means for associating each client with one of said versions (*paragraphs [0009] and [0026]; Fig. 7, step 706; paragraphs [0039]-[0041]*); and means for directing, using said JNDI proxy, the version associated with a particular client to said particular client upon a

request by said particular client for said J2EE program (*Fig. 7, steps 710-714; paragraphs [0031-0032]; paragraphs [0039]-[0041]*).

Claim 15: A computer program product for managing the invocation of multiple versions of a J2EE program, stored on an application server, among multiple clients accessing the application server, using an identical service name for the invocation of the multiple versions of the J2EE program (*paragraph [0009]*), the computer program product comprising a computer-readable storage medium having computer-readable program code embodied in the medium, the computer-readable program code comprising: computer-readable program code that interposes a JNDI proxy between each client and the application server (*paragraph [0009]; Fig. 3*); computer-readable program code that associates each client with one of said versions (*paragraphs [0009] and [0026]; Fig. 7, step 706; paragraphs [0039]-[0041]*); and computer-readable program code that directs, using said JNDI proxy, the version associated with a particular client to said particular client upon a request by said particular client for said J2EE program (*Fig. 7, steps 710-714; paragraphs [0031-0032]; paragraphs [0039]-[0041]*).

Claim 22: A system for managing the invocation of multiple versions of a J2EE program, stored on an application server, among multiple clients accessing the application server (*paragraph [0009]*), comprising: storing said multiple versions of said J2EE program on a single application server (*paragraphs [0025]-[0026]; Figs. 3-6*); and selectively serving said multiple versions of said J2EE program to said clients based upon a version specified by said client (*paragraph [0025]-[0026]; Figs. 3-6*).

The present invention teaches a method, system, and computer program product enabling multiple versions/releases of a J2EE application to be served to clients from a single application server, *using* an identical *service name* to invoke the multiple versions. One or more JNDI proxy servers ("JNDI proxies") are situated between each client and the application server.

Specifically, claim 1 recites "interposing a Java Naming and Directory Interface (JNDI) proxy between each client and the application server" (line 4). The JNDI proxies allow the identical public "service name" to be utilized by different clients to access different versions of programs/services on the application server. Claim 1 further recites "associating each client with one of said versions; and using said JNDI proxy, directing the version associated with a particular client to said particular client upon a request by said particular client for said J2EE program" (lines 5-7). The JNDI proxies perform this association by translating the service name into a non-public "alias name" on behalf of the client. The alias name is a private name that the service provider who administers the application server understands and uses to locate the specific version of programs/services that the clients need.

Independent claims 8 and 15 contain essentially identical elements in system and computer program product form, respectively.

Claim 22 is focused on a system for managing the invocation of multiple versions of a J2EE program, in which the multiple versions are stored on a single J2EE server and are selectively served to clients based on specification of a version by the client.

6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Applicant requests the Board to review the following rejection:

1. Rejection of claims 1-2, 4-9, 11-16, and 18-21 under 35 U.S.C. §102(e) as being anticipated by U.S. Patent Application Publication No. 2004/0168169 to Ebro et al.
2. Rejection of claim 22 under 35 U.S.C. §102(e) as being anticipated by U.S. Patent Application Publication No. 2004/0168169 to Ebro et al.

7. ARGUMENT

The MPEP and case law provide the following definition of anticipation for the purposes of 35 U.S.C. §102:

“A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.”
MPEP §2131 citing *Verdegaal Bros. v. Union Oil Company of California*, 814 F.2d 628, 631, 2 U.S.P.Q. 2d 1051, 1053 (Fed. Cir. 1987)

1. Rejection of Claims 1-2, 4-9, 11-16, and 18-21 under 35 U.S.C. §102(e)

As noted above, the present claimed invention as claimed in claims 1-2, 4-9, 11-16 and 18-21 is directed to version management of a J2EE program, and includes associating each client with one of multiple versions of a J2EE program stored on an application server based upon a request by the particular client. A JNDI proxy is interposed between *each* client and the application server, and the proxies direct the appropriate version to the requesting client. An identical service name is used for the invocation of the multiple versions of the J2EE program.

As a preliminary matter, the Examiner asserts that Ebro anticipates the claimed invention, and then points to specific paragraphs of Ebro while in many cases not indicating which aspects of the paragraph are analogous to the claim elements of the claimed invention. As an example, in the Advisory Action mailed on December 24, 2008, in response to the argument presented by the Applicant that Ebro does not appear to discuss the interposing of a JNDI proxy between each client and the application server, the Examiner states "In response to Applicant's argument, see Figure 10 and paragraphs [0190]-[0195] of [Ebro]." Applicant submits that nowhere in either Figure 10 nor in the cited paragraphs is there any mention of a JNDI proxy nor anything that could be reasonably interpreted as such; if the Examiner asserts that there is something analogous to a JNDI proxy server discussed in Ebro, it would assist in moving this case forward if the Examiner would point the Applicant directly to the analogous elements. Failure to do this makes it very difficult to properly address the Examiner's application of Ebro to the claimed invention.

Moving to the substantive arguments, the Examiner appears to be asserting that the use of "one domain name per DAJP" (paragraph [0190] of Ebro) is somehow the same as the use of identical service names for the invocation of multiple versions of the J2EE program as is claimed in independent claims 1, 8 and 15. This is incorrect. First, a "domain name" and a "service name" are two very different concepts as is well known. Second, in Ebro there is one domain name *per DAJP*; in other words, for each different DAJP, there is a different domain name used. In the case of the present invention, the *same* service name is used for different versions of the same program, i.e., the same JNDI service name is used for two or more different programs, albeit different versions of the same program.

Further, nothing in Ebro suggests, let alone teaches, that it can manage the invocation of multiple versions of a J2EE program as is claimed herein in claims 1, 8 and 15. In addition, there does not appear to be any discussion in Ebro of the interposing of a JNDI proxy between each client and the application server, as is claimed in claims 1, 8 and 15.

A typical prior art J2EE program has multiple levels, some of which are run on a client side, and some of which are run on the server side. In prior art systems, to avoid incompatibility issues, both the server side and the client side must be running the same version of the J2EE program.

In the claimed invention and claims in claims 1, 8 and 15, multiple versions of J2EE programs are maintained on the same server for assuring compatibility with all users, and all users use the same service name to invoke a different version of the program. This is advantageous over the prior art which either forced a client to upgrade its program to the current version, or redirected a client to another server which had the outdated version. Forcing a client to upgrade has inherent risks, e.g., the client's computer may not have the hardware resources to handle the upgraded version. The other alternative, redirecting clients to a second server with a matching software version, incurs additional hardware costs and requires additional programming effort to configure multiple application servers. For these reasons, prior art application providers do not attempt to provide multiple versions or releases of the same application. The present invention, however, patentably defines as novel over the prior art, including Ebro, by teaching associating each client with one of multiple versions of a J2EE

program stored on one application server, and allowing the multiple versions to be invoked using the same service name.

Without a teaching of associating each client with one of multiple versions of a J2EE program stored on one application server based upon a request using an identical service name by each client, or the interposing of a JNDI proxy between each client and the application server, Ebro cannot be said to anticipate the present invention. Accordingly, each of independent claims 1, 8 and 15, and all claims depending therefrom, patentably define over Ebro and are in condition for allowance.

For the foregoing reasons, applicants respectfully request this Board to overrule the Examiner's rejections and allow claims 1-2, 4-9, 11-16, and 18-21.

2. Rejection of Claim 22 under 35 U.S.C. §102(e)

In multiple rejections throughout the prosecution of the present application, the Examiner has included claim 22 as being rejected under 35 U.S.C. §102, indicating anticipation by Ebro. However, nowhere in the prosecution of the present application has the Examiner pointed out the aspects of Ebro that anticipate claim 22; instead, the Examiner has limited the explanation of the rejection to claims 1-21.

Notwithstanding this lack of explanation, applicant submits that Ebro neither teaches nor suggests the elements of claim 22. More particularly, claim 22 claims a system in which multiple versions of a J2EE program are stored on a single application server, and the multiple versions

are selectively served to clients based on a version specified by the client. Ebro neither teaches nor suggests these features.

Accordingly, for the foregoing reasons, applicants respectfully request this Board to overrule the Examiner's rejection and allow claim 22.

Respectfully submitted,

Date: August 24, 2009

/Mark D. Simpson/
Mark D. Simpson
Registration No. 32,942

SAUL EWING LLP
Centre Square West
1500 Market Street, 38th Floor
Philadelphia, PA 19102-2189
Telephone: 215 972 7880
Facsimile: 215 972 4169
Email: MSimpson@saul.com

CLAIMS APPENDIX

CLAIMS INVOLVED IN THIS APPEAL:

1. A method for managing the invocation of multiple versions of a J2EE program, stored on an application server, among multiple clients accessing the application server, using an identical service name for the invocation of the multiple versions of the J2EE program, comprising: interposing a JNDI proxy between each client and the application server; associating each client with one of said versions; and using said JNDI proxy, directing the version associated with a particular client to said particular client upon a request by said particular client for said J2EE program.

2. The method of claim 1, wherein associating each client with one of said versions comprises: assigning an identical service name used by each client to access said J2EE program; assigning an alias name for each version of said J2EE program; and for each client, associating said identical service name with the version of said J2EE program to be used by each of said clients.

4. The method of claim 2, wherein said J2EE program comprises one or more EJBs.

5. The method of claim 1, wherein said J2EE program comprises at least one JMS resource.

6. The method of claim 1, wherein said J2EE program comprises at least one JDBC datasource.

7. The method of claim 1, wherein said J2EE program is a system-oriented J2EE program.

8. A system for managing the invocation of multiple versions of a J2EE program, stored on an application server, among multiple clients accessing the application server, using an identical service name for the invocation of the multiple versions of the J2EE program, comprising: a JNDI proxy interposed between each client and the application server; means for associating each client with one of said versions; and means for directing, using said JNDI proxy, the version associated with a particular client to said particular client upon a request by said particular client for said J2EE program.

9. The system of claim 8, wherein said means for associating each client with one of said versions comprises: means for assigning an identical service name used by each client to access said J2EE program; means for assigning an alias name for each version of said J2EE program; and for each client, means for associating said identical service name with the version of said J2EE program to be used by each of said clients.

11. The system of claim 9, wherein said J2EE program comprises one or more EJBs.

12. The system of claim 8, wherein said J2EE program comprises at least one JMS resource.

13. The system of claim 8, wherein said J2EE program comprises at least one JDBC datasource.

14. The system of claim 8, wherein said J2EE program is a system-oriented J2EE program.

15. A computer program product for managing the invocation of multiple versions of a J2EE program, stored on an application server, among multiple clients accessing the application server, using an identical service name for the invocation of the multiple versions of the J2EE program, the computer program product comprising a computer-readable storage medium having computer-readable program code embodied in the medium, the computer-readable program code comprising: computer-readable program code that interposes a JNDI proxy between each client and the application server; computer-readable program code that associates each client with one of said versions; and computer-readable program code that directs, using said JNDI proxy, the version associated with a particular client to said particular client upon a request by said particular client for said J2EE program.

16. The computer program product of claim 15, wherein said computer readable program code that associates each client with one of said versions comprises: computer-readable program code that assigns an identical service name used by each client to access said J2EE program; computer-readable program code that assigns an alias name for each version of said J2EE program;

and computer-readable program code that for each client, associates said identical service name with the version of said J2EE program to be used by each of said clients.

18. The computer program product of claim 16, wherein said J2EE program comprises one or more EJBs.

19. The computer program product of claim 15, wherein said J2EE program comprises at least one JMS resource.

20. The computer program product of claim 15, wherein said J2EE program comprises at least one JDBC datasource.

21. The computer program product of claim 15, wherein said J2EE program is a system-oriented J2EE program.

22. A system for managing the invocation of multiple versions of a J2EE program, stored on an application server, among multiple clients accessing the application server, comprising: storing said multiple versions of said J2EE program on a single application server; and selectively serving said multiple versions of said J2EE program to said clients based upon a version specified by said client.

EVIDENCE APPENDIX

No additional evidence is presented.

RELATED PROCEEDINGS APPENDIX

No related proceedings are presented.